

[illegible]


```
0001 0 %TITLE 'VAX-11 CONVERT'
0002 0 MODULE CONV$SORT ( IDENT='V04-000',
0003 0                      OPTLEVEL=3
0004 0                      ) =
0005 0
0006 1 BEGIN
0007 1
0008 1 !*****
0009 1 !
0010 1 !  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 !  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 !  ALL RIGHTS RESERVED.
0013 1 !
0014 1 !  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 !  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 !  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 !  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 !  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 !  TRANSFERRED.
0020 1 !
0021 1 !  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 !  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 !  CORPORATION.
0024 1 !
0025 1 !  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 !  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 !
0028 1 !
0029 1 !*****
```

```
31 0030 1 ++
32 0031 1
33 0032 1 Facility: VAX-11 CONVERT
34 0033 1
35 0034 1 Abstract: CONVERT routines wich sort the input file on the output
36 0035 1 files primary key and to sort the output file by it's
37 0036 1 secondary key
38 0037 1
39 0038 1 Contents:
40 0039 1 SORT_PRIMARY
41 0040 1 SORT_SECONDARY
42 0041 1 SET_OP_SORT
43 0042 1
44 0043 1 Environment:
45 0044 1
46 0045 1 VAX/VMS Operating System
47 0046 1
48 0047 1 --
49 0048 1
50 0049 1
51 0050 1 Author: Keith B Thompson Creation date: August-1980
52 0051 1
53 0052 1
54 0053 1 Modified by:
55 0054 1
56 0055 1 V03-013 RAS0272 Ron Schaefer 16-Mar-1984
57 0056 1 Allow CONVERT to fastload/sort network files, now that
58 0057 1 SORT-32 can handle them.
59 0058 1
60 0059 1 V03-012 RAS0260 Ron Schaefer 6-Mar-1984
61 0060 1 Modify input file specs for SORT for LIB$FIND_FILE.
62 0061 1
63 0062 1 V03-011 KBT0502 Keith B. Thompson 19-Apr-1983
64 0063 1 Remove reference to SOR$M_SIGNAL
65 0064 1
66 0065 1 V03-010 KBT0467 Keith B. Thompson 21-Jan-1983
67 0066 1 Don't bother calling set_key_desc in sort_primary because
68 0067 1 we don't know if the file is still open for block io and
69 0068 1 set_key_desc does a $read. Also use the new sort interface.
70 0069 1
71 0070 1 V03-009 KBT0426 Keith B. Thompson 30-Nov-1982
72 0071 1 Fix a naming problem with the convert termorary file
73 0072 1 and remove sort error routine to get ready for the new
74 0073 1 sort interface which will signal errors.
75 0074 1
76 0075 1 V03-008 KBT0393 Keith B. Thompson 29-Oct-1982
77 0076 1 Use new set_key_desc routine
78 0077 1
79 0078 1 V03-007 KBT0379 Keith B. Thompson 21-Oct-1982
80 0079 1 Fix the linkage definition to set_key_block
81 0080 1
82 0081 1 V03-006 KBT0348 Keith B. Thompson 4-Oct-1982
83 0082 1 Use new linkage definitions (and fix history error
84 0083 1 in cwh0001!)
85 0084 1
86 0085 1 V03-005 CWH0001 CW Hobbs 17-Aug-1982
87 0086 1 Fix a history error in the last packet.
```


CONVSORT
V04-000

VAX-11 CONVERT

D 7
15-Sep-1984 23:48:01
14-Sep-1984 12:14:02

VAX-11 Bliss-32 V4.0-742
[CONV.SRC]CONVSORT.B32;1

Page 3
(2)

: 88
: 89
: 90
: 91
: 92
: 93
: 94
: 95
: 96
: 97
: 98
: 99
: 100
: 101
: 102
: 0087 1
: 0088 1
: 0089 1
: 0090 1
: 0091 1
: 0092 1
: 0093 1
: 0094 1
: 0095 1
: 0096 1
: 0097 1
: 0098 1
: 0099 1
: 0100 1
: 0101 1 :****

V03-004 KBT0125 Keith B. Thompson 10-Aug-1982
Get the file name length from RSL not RSS
V03-003 KBT0045 Keith Thompson 9-Apr-1982
Correct the way packed decimal sizes are given to sort
Also fix when we do stable sorts ie. only with dups
V03-002 KBT0027 Keith Thompson 30-Mar-1982
Chain the sort error messages
V03-001 KBT0014 Keith Thompson 17-Mar-1982
Pass sort a lrl so it will not choke on sys\$input

```
104 0102 1
105 0103 1 PSECT
106 0104 1      OWN      = _CONVSOWN      (PIC),
107 0105 1      GLOBAL  = _CONV$GLOBAL  (PIC),
108 0106 1      PLIT    = _CONV$PLIT    (SHARE,PIC),
109 0107 1      CODE    = _CONV$CODE    (SHARE,PIC);
110 0108 1
111 0109 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';
112 0110 1 LIBRARY 'SRCS:CONVERT';
113 0111 1
114 0112 1 FORWARD ROUTINE
115 0113 1      CONV$$SORT_PRIMARY : CL$SORT_PRIMARY,
116 0114 1      CONV$$SORT_SECONDARY : CL$SORT_SECONDARY,
117 0115 1      SET_UP_SORT : CL$JSB_REG_11 NOVALUE;
118 0116 1
119 0117 1 DEFINE_ERROR_CODES;
120 0118 1
121 0119 1 EXTERNAL ROUTINE
122 0120 1      CONV$$GET_VM : CL$GET_VM,
123 0121 1      CONV$$OPEN_IN,
124 0122 1      CONV$$RMS_OPEN_ERROR,
125 0123 1      CONV$$SET_KEY_DESC : CL$SET_KEY_DESC,
126 0124 1      CONV$$SEARCH_FILE,
127 0125 1      LIB$PUT_OUTPUT : ADDRESSING_MODE(GENERAL),
128 0126 1      SOR$BEGIN_SORT : ADDRESSING_MODE(GENERAL),
129 0127 1      SOR$PASS_FILES : ADDRESSING_MODE(GENERAL),
130 0128 1      SOR$SORT_MERGE : ADDRESSING_MODE(GENERAL),
131 0129 1      SOR$END_SORT : ADDRESSING_MODE(GENERAL);
132 0130 1
133 0131 1 EXTERNAL
134 0132 1      CONV$GL_SORT : LONG,
135 0133 1      CONV$GL_WORK_F : LONG,
136 0134 1
137 0135 1      CONV$AB_FLAGS : BLOCK [ ,BYTE ],
138 0136 1
139 0137 1      CONV$AR_OUT_FILE_NAM : REF DESC_BLK, ! Output File
140 0138 1      CONV$GB_CURRENT_FILE : BYTE,
141 0139 1      CONV$GL_FILE_COUNT,
142 0140 1      CONV$AR_PROLOGUE,
143 0141 1      CONV$GW_MAX_REC_SIZ : WORD,
144 0142 1
145 0143 1      CONV$AB_IN_NAM : $NAM_DECL,
146 0144 1      CONV$AB_IN_FAB : $FAB_DECL,
147 0145 1      CONV$AB_IN_RAB : $RAB_DECL,
148 0146 1      CONV$AB_OUT_NAM : $NAM_DECL,
149 0147 1      CONV$AB_OUT_FAB : $FAB_DECL,
150 0148 1      CONV$AB_OUT_RAB : $RAB_DECL;
151 0149 1
152 0150 1 EXTERNAL LITERAL
153 0151 1      SOR$M_STABLE,
154 0152 1      SOR$GR_RECORD,
155 0153 1      SOR$GK_ADDRESS,
156 0154 1      SOR$GK_INDEX;
157 0155 1
158 0156 1 ! SORT Temporary File Name Data
159 0157 1 !
160 0158 1 BIND
```



```
: 161      0159 1      CONV_TMP_STR      = UPLIT ('CONVWORK'),      ! Convert Temp. File Name
: 162      0160 1      CONV_DEF_STR      = UPLIT ('SYSS$SCRATCH:.TMP');      ! Default name
: 163      0161 1
: 164      0162 1      LITERAL
: 165      0163 1      CONV_TMP_SIZ = 8,
: 166      0164 1      CONV_DEF_SIZ = 16;
: 167      0165 1
: 168      0166 1      OWN
: 169      0167 1      CONV_TMP_DESC      : DESC_BLK,      ! Convert temp. file desc.
: 170      0168 1      TEMP_DESC      : DESC_BLK,      ! Expanded input file desc
: 171      0169 1
: 172      0170 1      ! Name block
: 173      0171 1
: 174      0172 1      RFA_NAM      : $NAM_DECL,      ! RFA Name Block
: 175      0173 1
: 176      0174 1      ! The fop bits are: Truncate eof - so to shrink file on multiple sorts
: 177      0175 1      ! Deferred write - of course
: 178      0176 1      ! Create if - We know sort is doing a create but we
: 179      0177 1      ! have created the file for him
: 180      0178 1
: 181      0179 1      FOP      : LONG INITIAL( FAB$M_TEF+FAB$M_DFW+FAB$M_CIF ),
: 182      0180 1      FILETYPE      : BYTE,
: 183      0181 1      RECORD_FMT      : BYTE,
: 184      0182 1      RECORDSIZ      : WORD;
: 185      0183 1
: 186      0184 1      GLOBAL
: 187      0185 1
: 188      0186 1      CONV$GL_RFA_BUFFER      : LONG,      ! Pointer to RFA Buffer
: 189      0187 1
: 190      0188 1      ! Work Files
: 191      0189 1
: 192      0190 1      CONV$AB_RFA_FAB      : $FAB_DECL,      ! RFA File FAB
: 193      0191 1
: 194      0192 1      CONV$AB_RFA_RAB      : $RAB_DECL;      ! RFA File RAB
: 195      0193 1
```

```
197 0194 1 %SBTTL 'INIT_SORT'
198 0195 1 ROUTINE INIT_SORT : NOVALUE =
199 0196 1 ++
200 0197 1
201 0198 1 Functional Description:
202 0199 1
203 0200 1     Initializes the rfa rms blocks which are used for sorting
204 0201 1
205 0202 1 Calling Sequence:
206 0203 1
207 0204 1     INIT_SORT()
208 0205 1
209 0206 1 Input Parameters:
210 0207 1     none
211 0208 1
212 0209 1 Implicit Inputs:
213 0210 1     none
214 0211 1
215 0212 1 Output Parameters:
216 0213 1     none
217 0214 1
218 0215 1 Implicit Outputs:
219 0216 1     none
220 0217 1
221 0218 1 Routines Called:
222 0219 1
223 0220 1     CONV$$GET_VM
224 0221 1
225 0222 1 Routine Value:
226 0223 1     none
227 0224 1
228 0225 1 Side Effects:
229 0226 1
230 0227 1     Clears the CONV$V_SORTINIT flag
231 0228 1
232 0229 1 --
233 0230 1
234 0231 2 BEGIN
235 0232 2
236 0233 2 LOCAL
237 0234 2     BYTES,
238 0235 2     VM_POINTER;
239 0236 2
240 0237 2     ! If sort has already been initialized then exit
241 0238 2
242 0239 2 IF NOT .CONV$AB_FLAGS [ CONV$V_SORTINIT ]
243 0240 2 THEN
244 0241 2     BEGIN
245 0242 2
246 0243 2     CONV$AB_FLAGS [ CONV$V_SORTINIT ] = _SET;
247 0244 2
248 0245 2     ! Allocate name block buffers and the rfa buffer
249 0246 2
250 0247 2     BYTES = ESA_BUF_SIZ + RSA_BUF_SIZ + RFA_BUF_SIZ;
251 0248 2
252 0249 2     CONV$GL_RFA_BUFFER = CONV$$GET_VM( .BYTES );
253 0250 2
```



```
254      VM_POINTER = .CONV$GL_RFA_BUFFER + RFA_BUF_SIZ;
255
256      ! Init the name block
257      !
258      $NAM_INIT ( NAM = RFA_NAM,
259                P  ESA = .VM_POINTER,
260                P  ESS = ESA_BUF_SIZ,
261                P  RSA = .VM_POINTER + ESA_BUF_SIZ,
262                P  RSS = RSA_BUF_SIZ );
263
264      ! Init the FAB
265      !
266      $FAB_INIT ( FAB = CONV$AB_RFA_FAB,
267                P  DNA = CONV_DEF_STR,
268                P  DNS = CONV_DEF_SIZ,
269                P  FAC = <BRO,GETS>,
270                P  FNA = CONV_TMP_STR,
271                P  FNS = CONV_TMP_SIZ,
272                P  FOP = <CBT,SQOS>,
273                P  NAM = RFA_NAM );
274
275      ! Init the RAB
276      !
277      $RAB_INIT ( RAB = CONV$AB_RFA_RAB,
278                P  FAB = CONV$AB_RFA_FAB,
279                P  ROP = BIO,
280                P  UBF = .CONV$GL_RFA_BUFFER,
281                P  USZ = RFA_BUF_SIZ );
282
283      END;
284
285      ! Set the record format and the record size
286      !
287      CONV$AB_RFA_FAB [ FAB$B_RFM ] = .RECORDFMT;
288      CONV$AB_RFA_FAB [ FAB$W_MRS ] = .RECORDSIZ;
289
290      ! Clear the delete flag so that we don't delete the temp file this time
291      !
292      CONV$AB_RFA_FAB [ FAB$V_DLT ] = _CLEAR;
293
294      ! Signal create error
295      !
296      CONV$AB_RFA_FAB [ FAB$SL_CTX ] = CONV$_CREA_ERR;
297
298      ! Create the file so that we get logical name direction to work and
299      ! we pass a good file name to sort
300      !
301      $CREATE( FAB=CONV$AB_RFA_FAB,ERR=CONV$$RMS_OPEN_ERROR );
302
303      $CLOSE( FAB=CONV$AB_RFA_FAB );
304
305      ! Set the delete flag so that we get rid of the temp file the next time
306      ! we open it
307      !
308      CONV$AB_RFA_FAB [ FAB$V_DLT ] = _SET;
309
310      ! Stuff the expanded file name into the temporary file descriptor
```

CONVSORT
V04-000

VAX-11 CONVERT
INIT_SORT

1 7
15-Sep-1984 23:48:01
14-Sep-1984 12:14:02

VAX-11 Bliss-32 V4.0-742
[CONV.SRC]CONVSORT.B32;1

Page 8
(4)

```

: 311      0308 2      !
: 312      0309 2      !
: 313      0310 2      !
: 314      0311 2      !
: 315      0312 2      !
: 316      0313 2      !
: 317      0314 1      !
                                CONV_TMP_DESC [ DSCSW_LENGTH ] = .RFA_NAM [ NAM$B_RSL ];
                                CONV_TMP_DESC [ DSCSA_POINTER ] = .RFA_NAM [ NAM$C_RSA ];
                                RETURN
                                END;
```

```

                                .TITLE CONV$SORT VAX-11 CONVERT
                                .IDENT \V04-000\
                                .PSECT _CONVSPLIT,NOWRT,NOEXE, SHR, PIC,2
4D 54 2E 3A 48 43 54 4B 52 4F 57 56 4E 4F 43 00000 P.AAA: .ASCII \CONVWORK\
                                00008 P.AAB: .ASCII \SYSS$SCRATCH:.TMP\
                                00017
                                .PSECT _CONVS$GLOBAL,NOEXE, PIC,2
                                00000 CONV$GL_RFA_BUFFER::
                                .BLKB 4
                                00004 CONV$AB_RFA_FAB::
                                .BLKB 80
                                00054 CONV$AB_RFA_RAB::
                                .BLKB 68
                                .PSECT _CONVS$OWN,NOEXE, PIC,2
                                00000 CONV_TMP_DESC:
                                .BLKB 8
                                00008 TEMP_DESC:
                                .BLKB 8
                                00010 RFA_NAM: .BLKB 96
                                00070 FOP: .LONG 301989920
                                00074 FILETYPE:
                                .BLKB 1
                                00075 RECORDFMT:
                                .BLKB 1
                                00076 RECORDSIZ:
                                .BLKB 2
                                CONV_TMP_STR= P.AAA
                                CONV_DEF_STR= P.AAB
                                $RMS_PTR= RFA_NAM
                                $RMS_PTR= CONV$AB_RFA_FAB
                                $RMS_PTR= CONV$AB_RFA_RAB
                                .EXTRN CONVERT$ FACILITY
                                .EXTRN CONV$ FAO MAX, CONV$ BADBLK
                                .EXTRN CONV$ BADLOGIC, CONV$ BADSORT
                                .EXTRN CONV$ CONQUAL, CONV$ CREATEDSTM
                                .EXTRN CONV$ CREA_ERR, CONV$ DELPRI
                                .EXTRN CONV$ DUP, CONV$ EXTN_ERR
                                .EXTRN CONV$ FATALEXC, CONV$ FILLIM
                                .EXTRN CONV$ IDX_LIM, CONV$ ILL_KEY
                                .EXTRN CONV$ ILL_VALUE
                                .EXTRN CONV$ INP_FILES
```



```
.EXTRN CONVS_INSVIRMEM
.EXTRN CONVS_INVBKT, CONVS_KEY
.EXTRN CONVS_KEYREF, CONVS_LOADIDX
.EXTRN CONVS_NARG, CONVS_NT
.EXTRN CONVS_NOKEY, CONVS_NOTIDX
.EXTRN CONVS_NOTSEQ, CONVS_NOWILD
.EXTRN CONVS_ORDER, CONVS_OPENEXC
.EXTRN CONVS_OPENIN, CONVS_OPENOUT
.EXTRN CONVS_PAD, CONVS_PLV
.EXTRN CONVS_PROERR, CONVS_PROL_WRT
.EXTRN CONVS_READERR, CONVS_RSK
.EXTRN CONVS_RSZ, CONVS_RTL
.EXTRN CONVS_RTS, CONVS_SEQ
.EXTRN CONVS_UDF_BKS, CONVS_UDF_BLK
.EXTRN CONVS_VFC, CONVS_WRITEERR
.EXTRN CONVS$GET_VM, CONVS$OPEN_IN
.EXTRN CONVS$RMS_OPEN_ERROR
.EXTRN CONVS$SET_KEY_DESC
.EXTRN CONVS$SEARCH_FILE
.EXTRN LIB$PUT_OUTPUT, SOR$BEGIN_SORT
.EXTRN SOR$PASS_FILES, SOR$SORT_MERGE
.EXTRN SOR$END_SORT, CONVS$GL_SORT
.EXTRN CONVS$GL_WORK_F, CONVS$AB_FLAGS
.EXTRN CONVS$AR_OUT_FILE_NAM
.EXTRN CONVS$GB_CURRENT_FILE
.EXTRN CONVS$GL_FILE_COUNT
.EXTRN CONVS$AR_PROLOGUE
.EXTRN CONVS$GW_MAX_REC_SIZ
.EXTRN CONVS$AB_IN_NAM, CONVS$AB_IN_FAB
.EXTRN CONVS$AB_IN_RAB, CONVS$AB_OUT_NAM
.EXTRN CONVS$AB_OUT_FAB
.EXTRN CONVS$AB_OUT_RAB
.EXTRN SOR$M_STABLE, SOR$GK_RECORD
.EXTRN SOR$GK_ADDRESS, SOR$GK_INDEX
.EXTRN SYSS$CREATE, SYSS$CLOSE
```

```
.PSECT _CONVS$CODE, NOWRT, SHR, PIC, 2
```

```
OFFC 00000 INIT_SORT:
```

		58	0000'	CF	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 0195
		57	0000'	CF	9E	00007	MOVAB	\$RMS_PTR, R8	:
03	0000G	CF		05	E1	0000C	MOVAB	\$RMS_PTR, R7	:
				0098	31	00012	BBC	#5, CONVS\$AB_FLAGS+2, 1\$: 0239
	0000G	CF		20	88	00015	BRW	2\$:
		50	06A0	8F	3C	0001A	BISB2	#32, CONVS\$AB_FLAGS+2	: 0243
				50	DD	0001F	MOVZWL	#1696, BYTES	: 0247
				0000G	30	00021	PUSHL	BYTES	: 0249
		5E		04	C0	00024	BSBW	CONVS\$GET_VM	:
	FC	A7		50	D0	00027	ADDL2	#4, SP	:
	FC	A7	00000600	8F	C1	0002B	MOVL	R0, CONVS\$GL_RFA_BUFFER	:
0060	8F	6E		00	2C	00034	ADDL3	#1536, CONVS\$GL_RFA_BUFFER, VM_POINTER	: 0251
				68		0003B	MOVCS	#0, (SP), #0, #96, \$RMS_PTR	: 0259
		68	6002	8F	B0	0003C	MOVW	#24578, \$RMS_PTR	:
	02	A8	50	8F	90	00041	MOVB	#80, \$RMS_PTR+2	:
	04	A8	50	A6	9E	00046	MOVAB	80(R6), \$RMS_PTR+4	:
	0A	A8	50	8F	90	0004B	MOVB	#80, \$RMS_PTR+10	:

CONVSORT
V04-000

VAX-11 CONVERT
INIT_SORT

K 7
15-Sep-1984 23:48:01
14-Sep-1984 12:14:02

VAX-11 Bliss-32 V4.0-742
[CONV.SRC]CONVSORT.B32;1

Page 10
(4)

0050	8F	00	0C	A8		56	D0	00050	MOVL	VM_POINTER, SRMS_PTR+12	:	
			6E	6E		00	2C	00054	MOVCS	#0, (SP), #0, #80, SRMS_PTR	:	0270
				67	5003	67	B0	0005B			:	
			04	A7	00200040	8F	D0	0005C	MOVW	#20483, SRMS_PTR	:	
			16	A7	42	8F	D0	00061	MOVL	#2097216, SRMS_PTR+4	:	
			1F	A7		8F	90	00069	MOVB	#66, SRMS_PTR+22	:	
			28	A7		02	90	0006E	MOVB	#2, SRMS_PTR+31	:	
			2C	A7	0000'	68	9E	00072	MOVAB	RFA_NAM, SRMS_PTR+40	:	
			30	A7	0000'	CF	9E	00076	MOVAB	CONV_TMP_STR, SRMS_PTR+44	:	
			34	A7	1008	CF	9E	0007C	MOVAB	CONV_DEF_STR, SRMS_PTR+48	:	
0044	8F	00		6E		8F	B0	00082	MOVW	#410Z, SRMS_PTR+52	:	
						00	2C	00088	MOVCS	#0, (SP), #0, #68, SRMS_PTR	:	0278
					50	A7		0008F			:	
			50	A7	4401	8F	B0	00091	MOVW	#17409, SRMS_PTR	:	
			54	A7	0800	8F	3C	00097	MOVZWL	#2048, SRMS_PTR+4	:	
			70	A7	0600	8F	B0	0009D	MOVW	#1536, SRMS_PTR+32	:	
			74	A7	FC	A7	D0	000A3	MOVL	CONVSGL_RFA_BUFFER, SRMS_PTR+36	:	
			008C	C7		67	9E	000AB	MOVAB	CONVSAB_RFA_FAB, SRMS_PTR+60	:	
			1F	A7	65	A8	90	000AD	MOVW	RECORDFMT, CONVSAB_RFA_FAB+31	:	0284
			36	A7	66	A8	B0	000B2	MOVW	RECORDSIZ, CONVSAB_RFA_FAB+54	:	0285
			05	A7	80	8F	8A	000B7	BICB2	#128, CONVSAB_RFA_FAB+5	:	0289
			18	A7	00000000G	8F	D0	000BC	MOVL	#CONVS_CREA_ERR, CONVSAB_RFA_FAB+24	:	0293
					0000G	CF	9F	000C4	PUSHAB	CONVS\$SRMS_OPEN_ERROR	:	0298
						57	DD	000C8	PUSHL	R7	:	
			00000000G	00		02	FB	000CA	CALLS	#2, SYSS\$CREATE	:	
						57	DD	000D1	PUSHL	R7	:	0300
			00000000G	00		01	FB	000D3	CALLS	#1, SYSS\$CLOSE	:	
			05	A7	80	8F	88	000DA	BISB2	#128, CONVSAB_RFA_FAB+5	:	0305
			F0	A8	03	A8	9B	000DF	MOVZBW	RFA_NAM+3, CONV_TMP_DESC	:	0309
			F4	A8	04	A8	D0	000E4	MOVL	RFA_NAM+4, CONV_TMP_DESC+4	:	0310
						04	000E9		RET		:	0314

; Routine Size: 234 bytes, Routine Base: _CONVS\$CODE + 0000


```
0315 1 %SBTTL 'SORT_PRIMARY'
0316 1 GLOBAL ROUTINE CONV$$SORT_PRIMARY : CL$SORT_PRIMARY =
0317 1 ++
0318 1
0319 1 Functional Description:
0320 1
0321 1     This routine will sort the input file, pointed to by in_fab, according
0322 1     to the primary key of the output file.
0323 1
0324 1 Calling Sequence:
0325 1
0326 1     CONV$$SORT_PRIMARY()
0327 1
0328 1 Input Parameters:
0329 1     none
0330 1
0331 1 Implicit Inputs:
0332 1
0333 1     input and output rms blocks
0334 1
0335 1 Output Parameters:
0336 1     none
0337 1
0338 1 Implicit Outputs:
0339 1     none
0340 1
0341 1 Routines Called:
0342 1
0343 1     INIT_SORT
0344 1     SOR$PASS_FILES
0345 1     SORT_ERROR
0346 1     CONV$$SEARCH_FILE
0347 1     SET_UP_SORT
0348 1     SOR$SORT_MERGE
0349 1     SOR$END_SORT
0350 1
0351 1 Routine Value:
0352 1
0353 1     Success of random errors
0354 1
0355 1 Side Effects:
0356 1
0357 1     Open the rfa file if CONV$V_RFA is set
0358 1
0359 1 --
0360 1
0361 2 BEGIN
0362 2
0363 2 DEFINE_KEY_DESC;
0364 2
0365 2 LOCAL
0366 2     IN_DEVICE      : BLOCK [ 1, LONG ],
0367 2     RFA            : LONG;
0368 2
0369 2     ! Set the key descriptor to key = 0 (always in prologue)
0370 2     !
0371 2 KEY_DESC = .CONV$AR_PROLOGUE;
```

```
376 0372 2
377 0373 2
378 0374 2
379 0375 2
380 0376 2
381 0377 2
382 0378 2
383 0379 2
384 0380 2
385 0381 2
386 0382 2
387 0383 2
388 0384 2
389 0385 2
390 0386 2
391 0387 2
392 0388 2
393 0389 2
394 0390 2
395 0391 2
396 0392 2
397 0393 2
398 0394 2
399 0395 2
400 0396 2
401 0397 2
402 0398 2
403 0399 2
404 0400 2
405 0401 2
406 0402 2
407 0403 2
408 0404 2
409 0405 2
410 0406 2
411 0407 2
412 0408 2
413 0409 2
414 0410 2
415 0411 2
416 0412 2
417 0413 2
418 0414 2
419 0415 2
420 0416 2
421 0417 2
422 0418 2
423 0419 2
424 0420 2
425 0421 2
426 0422 2
427 0423 2
428 0424 2
429 0425 2
430 0426 2
431 0427 2
432 0428 2

! If the input file is open close it
!
IF .CONVSAB_FLAGS [ CONVSV_IN ]
THEN
  BEGIN
    $DISCONNECT( RAB=CONVSAB_IN_RAB );
    $CLOSE( FAB=CONVSAB_IN_FAB );
    CONVSAB_FLAGS [ CONVSV_IN ] = _CLEAR
  END;

IN_DEVICE = .CONVSAB_IN_FAB [ FAB$L_DEV ];

! If the device char. are zero (process perminant files) or
! if the input file is not from disk or
! it is a record oriented device (terminals) or
! it's a network file or
! it's a terminal (be reduntandt) or
! there is more than one input file
! then we do a normal record sort otherwise we do a RFA sort to save time
IF ( .IN_DEVICE EQLU 0 ) OR
  .IN_DEVICE [ DEV$V_SQD ] OR
  .IN_DEVICE [ DEV$V_NET ] OR
  .IN_DEVICE [ DEV$V_REC ] OR
  .IN_DEVICE [ DEV$V_TRM ] OR
  ( .CONVSGL_FILE_COUNT GTR 1 )
THEN
  BEGIN
    RFA = _CLEAR;
    RECORDFMT = FAB$C_VAR;
    RECORDSIZ = 0;
  END
ELSE
  BEGIN
    RFA = _SET;
    RECORDFMT = FAB$C_FIX;
    RECORDSIZ = 6;
  END;

! Initialize the RMS blocks used in the sort
!
INIT_SORT();

! Pass the file names 1st input and output
!
TEMP_DESC [ DSC$W_LENGTH ] = .CONVSAB_IN_FAB [ FAB$B_FNS ];
TEMP_DESC [ DSC$A_POINTER ] = .CONVSAB_IN_FAB [ FAB$C_FNA ];

SOR$PASS_FILES( TEMP_DESC,
  CONV-TMP_DESC,
  FILETYPE,
  RECORDFMT,
  0,
  0,
  0,
  0,
```



```

433      FOP );
434
435      CONV$GB_CURRENT_FILE = 1;
436
437      ! Pass the rest of the input names
438      !
439      UNTIL .CONV$GB_CURRENT_FILE GTR ( .CONV$GL_FILE_COUNT - 1 )
440      DO
441      BEGIN
442
443          ! Parse and search for the file (This uses the IN_FAB and IN_NAM
444          ! since they are not used again)
445          !
446          RET_ON_ERROR( CONV$$SEARCH_FILE() );
447
448          ! Pass the file spec
449          !
450          TEMP_DESC [ DSC$W_LENGTH ] = .CONV$AB_IN_FAB [ FAB$B_FNS ];
451          TEMP_DESC [ DSC$A_POINTER ] = .CONV$AB_IN_FAB [ FAB$B_FNA ];
452
453          SOR$PASS_FILES( TEMP_DESC );
454
455          CONV$GB_CURRENT_FILE = .CONV$GB_CURRENT_FILE + 1
456
457      END;
458
459      ! If useing rfa file as index file do an index sort else do record sort
460      !
461      IF .RFA
462      THEN
463          SET_UP_SORT( SOR$GK_ADDRESS )
464      ELSE
465          SET_UP_SORT( SOR$GK_RECORD );
466
467      ! Do the sort
468      !
469      SOR$SORT_MERGE();
470
471      SOR$END_SORT();
472
473      ! Reopen the correct input files
474      !
475      IF .RFA
476      THEN
477      BEGIN
478
479          ! OPEN the input file and the new RFA file
480          !
481          RET_ON_ERROR( CONV$$OPEN_IN() );
482
483          ! Connect the additional file containing the RFAs pointing th the real
484          ! file
485          !
486          $OPEN( FAB=CONV$AB_RFA_FAB );
487          $CONNECT( RAB=CONV$AB_RFA_RAB );
488
489          CONV$AB_FLAGS [ CONV$V_RFA ] = _SET;
```

```

: 490      0486      3
: 491      0487
: 492      0488      ! Set access to the real input file to RFA
: 493      0489      ! CONVSAB_IN_RAB [ RAB$B_RAC ] = RAB$C_RFA
: 494      0490
: 495      0491      END
: 496      0492      ELSE
: 497      0493
: 498      0494      ! OPEN the sorted file as if it was the input file
: 499      0495
: 500      0496      BEGIN
: 501      0497
: 502      0498      ! The real input RAB points to the RFA FAB
: 503      0499
: 504      0500      CONVSAB_IN_RAB [ RAB$L_FAB ] = CONVSAB_RFA_FAB;
: 505      0501
: 506      0502      ! Open the RFA fab which is the new sorted input file NOTE: This is
: 507      0503      ! not a file of RFAs an above
: 508      0504
: 509      0505      $OPEN( FAB=CONVSAB_RFA_FAB );
: 510      0506      $CONNECT( RAB=CONVSAB_IN_RAB );
: 511      0507
: 512      0508      CONVSAB_FLAGS [ CONVS$V_SOR ] = _SET
: 513      0509
: 514      0510      END;
: 515      0511
: 516      0512      ! Since it only makes sence to sort once
: 517      0513
: 518      0514      CONVS$GL_SORT = _CLEAR;
: 519      0515
: 520      0516      RETURN CONVS$_SUCCESS
: 521      0517
: 522      0518      END;

```

```

                                .EXTRN  SYSSDISCONNECT, SYSSOPEN
                                .EXTRN  SYSSCONNECT
                                52 DD 00000 CONVS$$SORT_PRIMARY::
                                PUSH    R2
                                5B      0000G CF D0 00002      MOVL    CONVSAR_PROLOGUE, KEY_DESC
                                1B      0000G CF E9 00007      BLBC    CONVSAB_FLAGS+2, 1$
                                00000000G 00      0000G CF 9F 0000C      PUSHAB  CONVSAB_IN_RAB
                                00000000G 00      0000G 01 FB 00010      CALLS   #1, SYSSDISCONNECT
                                0000G      00      0000G CF 9F 00017      PUSHAB  CONVSAB_IN_FAB
                                0000G      00      0000G 01 FB 0001B      CALLS   #1, SYSSCLOSE
                                0000G      00      0000G 01 8A 00022      BICB2   #1, CONVSAB_FLAGS+2
                                50      0000G CF D0 00027 1$:      MOVL    CONVSAB_IN_FAB+64, IN_DEVICE
                                16      13 0002C      BEQL    2$
                                12      50      05 E0 0002E      BBS     #5, IN_DEVICE, 2$
                                OE      50      0D E0 00032      BBS     #13, IN_DEVICE, 2$
                                07      0B      50 E8 00036      BLBS    IN_DEVICE, 2$
                                01      0000G 02 E0 00039      BBS     #2, IN_DEVICE, 2$
                                0000G      00      0000G CF D1 0003D      CMPL    CONVS$C_FILE_COUNT, #1
                                0D      15 00042      BLEQ    3$
                                52      D4 00044 2$:      CLRL    RFA

```

0316
0371
0375
0378
0379
0380
0383
0393
0394
0395
0396
0397
0398
0401

0000'	CF	0000'	02	90	00046	MOVB	#2, RECORDFMT	0402	
		0000'	CF	B4	0004B	CLRW	RECORDSIZ	0403	
			0D	11	0004F	BRB	4\$	0393	
	52		01	D0	00051	3\$:	MOVL	#1, RFA	0407
0000'	CF		01	90	00054	MOVB	#1, RECORDFMT	0408	
0000'	CF		06	B0	00059	MOVW	#6, RECORDSIZ	0409	
FEB3	CF		00	FB	0005E	4\$:	CALLS	#0, INIT_SORT	0414
0000'	CF	0000G	CF	9B	00063	MOVZBW	CONVSAB_IN_FAB+52, TEMP_DESC	0418	
0000'	CF	0000G	CF	D0	0006A	MOVL	CONVSAB_IN_FAB+44, TEMP_DESC+4	0419	
		0000'	CF	9F	00071	PUSHAB	FOP	0421	
			7E	7C	00075	CLRQ	-(SP)		
			7E	7C	00077	CLRQ	-(SP)		
		0000'	CF	9F	00079	PUSHAB	RECORDFMT		
0000'	CF		CF	9F	0007D	PUSHAB	FILETYPE		
0000'	CF		CF	9F	00081	PUSHAB	CONV_TMP_DESC		
0000'	CF		CF	9F	00085	PUSHAB	TEMP_DESC		
00000000G	00		09	FB	00089	CALLS	#9, SOR\$PASS_FILES		
0000G	CF		01	90	00090	MOVB	#1, CONVSGB_CURRENT_FILE	0431	
0000G	CF		01	C3	00095	5\$:	SUBL3	#1, CONVSGL_FILE_COUNT, R0	0435
50	0000G	50	08	00	ED	CMPZV	#0, #8, CONVSGB_CURRENT_FILE, R0		
				27	14	BGTR	6\$		
0000G	CF		00	FB	000A4	CALLS	#0, CONVS\$SEARCH_FILE	0442	
	4C		50	E9	000A9	BLBC	STATUS, 9\$		
0000'	CF	0000G	CF	9B	000AC	MOVZBW	CONVSAB_IN_FAB+52, TEMP_DESC	0446	
0000'	CF	0000G	CF	D0	000B3	MOVL	CONVSAB_IN_FAB+44, TEMP_DESC+4	0447	
		0000'	CF	9F	000BA	PUSHAB	TEMP_DESC	0449	
00000000G	00		01	FB	000BE	CALLS	#1, SOR\$PASS_FILES		
		0000G	CF	96	000C5	INCB	CONVSGB_CURRENT_FILE	0451	
			CA	11	000C9	BRB	5\$		
		08	52	E9	000CB	6\$:	BLBC	RFA, 7\$	0457
		00000000G	8F	DD	000CE	PUSHL	#SOR\$GK_ADDRESS	0459	
			06	11	000D4	BRB	8\$		
		00000000G	8F	DD	000D6	7\$:	PUSHL	#SOR\$GK_RECORD	0461
		0000V	30	000DC	8\$:	BSBW	SET_UP_SORT		
			04	C0	000DF	ADDL2	#4, SP		
00000000G	00		00	FB	000E2	CALLS	#0, SOR\$SORT_MERGE	0465	
00000000G	00		00	FB	000E9	CALLS	#0, SOR\$END_SORT	0467	
	2A		52	E9	000F0	BLBC	RFA, 10\$	0471	
0000G	CF		00	FB	000F3	CALLS	#0, CONVS\$OPEN_IN	0477	
	4B		50	E9	000F8	9\$:	BLBC	STATUS, 12\$	
		0000'	CF	9F	000FB	PUSHAB	CONVSAB_RFA_FAB	0482	
00000000G	00		01	FB	000FF	CALLS	#1, SYS\$OPEN		
		0000'	CF	9F	00106	PUSHAB	CONVSAB_RFA_RAB	0483	
00000000G	00		01	FB	0010A	CALLS	#1, SYS\$CONNECT		
0000G	CF		10	88	00111	BISB2	#16, CONVSAB_FLAGS+2	0485	
0000G	CF		02	90	00116	MOVB	#2, CONVSAB_IN_RAB+30	0489	
			22	11	0011B	BRB	11\$		
0000G	CF	0000'	CF	9E	0011D	10\$:	MOVAB	CONVSAB_RFA_FAB, CONVSAB_IN_RAB+60	0500
		0000'	CF	9F	00124	PUSHAB	CONVSAB_RFA_FAB	0505	
00000000G	00		01	FB	00128	CALLS	#1, SYS\$OPEN		
		0000G	CF	9F	0012F	PUSHAB	CONVSAB_IN_RAB	0506	
00000000G	00		01	FB	00133	CALLS	#1, SYS\$CONNECT		
0000G	CF		08	88	0013A	BISB2	#8, CONVSAB_FLAGS+2	0508	
		0000G	CF	D4	0013F	11\$:	CLRL	CONVSGL_SORT	0514
			01	D0	00143	MOVL	#1, R0	0516	
			04	BA	00146	12\$:	POPR	#^M<R2>	0518
			05	00148	RSB				

CONV\$SORT
V04-000

VAX-11 CONVERT
SORT_PRIMARY

D 8
15-Sep-1984 23:48:01
14-Sep-1984 12:14:02

VAX-11 Bliss-32 V4.0-742
[CONV.SRC]CONV\$SORT.B32;1

Page 16
(5)

; Routine Size: 329 bytes, Routine Base: _CONV\$CODE + 00EA


```
524 0519 1 %SBTTL 'SORT_SECONDARY'
525 0520 1 GLOBAL ROUTINE CONV$$$SORT_SECONDARY : CL$SORT_SECONDARY =
526 0521 1 ++
527 0522 1
528 0523 1 Functional Description:
529 0524 1
530 0525 1 This routine will sort the OUTPUT file according to a specified
531 0526 1 key of the OUTPUT file.
532 0527 1
533 0528 1 Calling Sequence:
534 0529 1
535 0530 1 CONV$$$SORT_SECONDARY()
536 0531 1
537 0532 1 Input Parameters:
538 0533 1 none
539 0534 1
540 0535 1 Implicit Inputs:
541 0536 1 none
542 0537 1
543 0538 1 Output Parameters:
544 0539 1 none
545 0540 1
546 0541 1 Implicit Outputs:
547 0542 1 none
548 0543 1
549 0544 1 Routines Called:
550 0545 1
551 0546 1 INIT SORT
552 0547 1 SOR$PASS FILES
553 0548 1 SET UP SORT
554 0549 1 SOR$SORT MERGE
555 0550 1 SOR$END_SORT
556 0551 1
557 0552 1 Routine Value:
558 0553 1
559 0554 1 Success or random errors
560 0555 1
561 0556 1 Side Effects:
562 0557 1
563 0558 1 Closes and reopens the output file
564 0559 1 Closes the rfa file if it was open then opens it
565 0560 1
566 0561 1 --
567 0562 1
568 0563 2 BEGIN
569 0564 2
570 0565 2 DEFINE_KEY_DESC:
571 0566 2
572 0567 2 ! If the RFA file was open close it. The file will be used as output of sort.
573 0568 2 !
574 0569 2 IF .CONV$AB_FLAGS [ CONV$V_RFA ]
575 0570 2 THEN
576 0571 3 BEGIN
577 0572 3 ERRCHK( $DISCONNECT( RAB=CONV$AB_RFA_RAB ) CONV$BADLOGIC );
578 0573 3 ERRCHK( $CLOSE( FAB=CONV$AB_RFA_FAB ) CONV$BADLOGIC );
579 0574 3
580 0575 3 CONV$AB_FLAGS [ CONV$V_RFA ] = _CLEAR;
```

```
581 0576 2
582 0577 3
583 0578 3
584 0579 4
585 0580 4
586 0581 2
587 0582 2
588 0583 2
589 0584 2
590 0585 2
591 0586 2
592 0587 2
593 0588 2
594 0589 2
595 0590 2
596 0591 2
597 0592 2
598 0593 2
599 0594 2
600 0595 2
601 0596 2
602 0597 2
603 0598 2
604 0599 2
605 0600 2
606 0601 2
607 0602 2
608 0603 2
609 0604 2
610 0605 2
611 0606 2
612 0607 2
613 0608 2
614 0609 2
615 0610 2
616 0611 2
617 0612 2
618 0613 2
619 0614 2
620 0615 2
621 0616 2
622 0617 2
623 0618 2
624 0619 2
625 0620 2
626 0621 2
627 0622 2
628 0623 2
629 0624 2
630 0625 2
631 0626 2
632 0627 2
633 0628 2
634 0629 2
635 0630 2
636 0631 2
637 0632 2

! Also remove the entry in the directory
SERASE( FAB=CONVSAB_RFA_FAB )

END;

! Secondary key sorts are always tag sorts therefore we need a var. file
RECORDFMT = FABSC_VAR;
RECORDSIZ = 0;

! Init sort if necc. and get a file name
INIT_SORT();

! To conserve space ect. use the RFA fab and rab therefore reset
! the RFA rab so we can do record I/O on it. We can use the rfa buffer
! since it is at least 512 bytes long and a key is only 256 + 6 byte rfa

! Clear the BIO flag
CONVSAB_RFA_RAB [ RAB$V_BIO ] = _CLEAR;

! Close the current output file so that SORT can get at it
$DISCONNECT( RAB=CONVSAB_OUT_RAB );
ERRCHK( $CLOSE( FAB=CONVSAB_OUT_FAB ),CONVS_BADLOGIC );

CONVSAB_FLAGS [ CONVS$V_OUT ] = _CLEAR;

! Pass the file names
! To avoid some file name problems pass the expanded string of the
! output file
TEMP_DESC [ DSCSW_LENGTH ] = .CONVSAB_OUT_NAM [ NAM$B_RSL ];
TEMP_DESC [ DSCSA_POINTER ] = .CONVSAB_OUT_NAM [ NAM$C_RSA ];

SOR$PASS_FILES( TEMP_DESC,
                CONV_TMP_DESC,
                FILETYPE,
                RECORDFMT,
                0,
                0,
                0,
                0,
                FOP );

! Get ready to do a index sort of the file
SET_UP_SORT( SOR$GK_INDEX );

! Start the sort and finish it
SOR$SORT_MERGE();
SOR$END_SORT();
```



```

: 638      0633 2
: 639      0634 2
: 640      0635 2
: 641      0636 2
: 642      0637 2
: 643      0638 2
: 644      0639 2
: 645      0640 2
: 646      0641 2
: 647      0642 2
: 648      0643 2
: 649      0644 2
: 650      0645 1

! ReOPEN the output file and the new RFA-INDEX file
$OPEN( FAB=CONVSAB OUT FAB );
$CONNECT( RAB=CONVSAB OUT RAB );
CONVSAB_FLAGS [ CONVSAB_OUT ] = _SET;

$OPEN( FAB=CONVSAB RFA FAB );
$CONNECT( RAB=CONVSAB RFA RAB );
CONVSAB_FLAGS [ CONVSAB_RFA ] = _SET;

RETURN SSS_NORMAL
END;
```

.EXTRN SYSSERASE

```

32      0000G CF      0000' 52 DD 00000 CONVSORT_SECONDARY::
      00000000G 00      01 FB 0000C PUSHB R2
      52      50 D0 00013 BBC #4, CONVSAB_FLAGS+2, 1$
      50      52 E9 00016 PUSHAB CONVSAB_RFA_RAB
      0000' 01 FB 0001D CALLS #1, SYSSDISCONNECT
      00000000G 00      50 D0 00024 MOVL R0, STATUS
      52      52 E9 00027 BLBC STATUS, 2$
      3F      10 8A 0002A PUSHAB CONVSAB_RFA_FAB
      0000G CF      0000' 01 FB 0002F CALLS #1, SYSSCLOSE
      00000000G 00      50 D0 0003A 1$: MOVB #2, RECORDFMT
      0000' CF      0000' 02 FB 0003F CLRW RECORDSIZ
      FD85 CF      00 FB 00043 CALLS #0, INIT_SORT
      0000' CF      08 8A 00048 BICB2 #8, CONVSAB_RFA_RAB+5
      00000000G 00      0000G CF 9F 0004D PUSHAB CONVSAB_OUT_RAB
      00000000G 00      01 FB 00051 CALLS #1, SYSSDISCONNECT
      00000000G 00      CF 9F 00058 PUSHAB CONVSAB_OUT_FAB
      52      01 FB 0005C CALLS #1, SYSSCLOSE
      13      50 D0 00063 MOVL R0, STATUS
      00000000G 00      52 E8 00066 BLBS STATUS, 3$
      00000000G 00      8F DD 00069 2$: PUSHB #CONVSAB_BADLOGIC
      50      01 FB 0006F CALLS #1, LIBSSIGNAL
      0085 31 00079 MOVL STATUS, R0
      0000G CF      02 8A 0007C 3$: BRW 4$
      0000' CF      0000G CF 9B 00081 BICB2 #2, CONVSAB_FLAGS+2
      0000' CF      0000G CF D0 00088 MOVZBW CONVSAB_OUT_NAM+3, TEMP_DESC
      0000' CF      0000' CF 9F 0008F MOVL CONVSAB_OUT_NAM+4, TEMP_DESC+4
      7E 7C 00093 PUSHAB FOP
      7E 7C 00095 CLRW -(SP)
      0000' CF 9F 00097 CLRW -(SP)
      0000' CF 9F 0009B PUSHAB RECORDFMT
      0000' CF 9F 0009F PUSHAB FILETYPE
      0000' CF 9F 000A3 PUSHAB CONV_TMP_DESC
      00000000G 00      09 FB 000A7 CALLS #9, SORSPASS_FILES
```

CONVSORT
V04-000

VAX-11 CONVERT
SORT_SECONDARY

H 8
15-Sep-1984 23:48:01
14-Sep-1984 12:14:02

VAX-11 Bliss-32 V4.0-742
[CONV.SRC]CONVSORT.B32;1

Page 20
(6)

00000000G	8F	DD	000AE	PUSHL	#SOR\$GK_INDEX	: 0627
00000000G	0000V	30	000B4	BSBW	SET_UP_SORT	:
00000000G	5E	04	C0	ADDL2	#4,-SP-	:
00000000G	00	00	FB	CALLS	#0, SOR\$SORT_MERGE	: 0631
00000000G	00	00	FB	CALLS	#0, SOR\$SEND_SORT	: 0632
00000000G	00	0000G	CF	PUSHAB	CONV\$AB_OUT_FAB	: 0636
00000000G	00	0000G	01	CALLS	#1, SYS\$OPEN	:
00000000G	00	0000G	CF	PUSHAB	CONV\$AB_OUT_RAB	: 0637
00000000G	00	0000G	01	CALLS	#1, SYS\$CONNECT	:
00000000G	CF	02	88	BISB2	#2, CONV\$AB_FLAGS+2	: 0638
00000000G	00	0000'	CF	PUSHAB	CONV\$AB_RFA_FAB	: 0640
00000000G	00	0000'	01	CALLS	#1, SYS\$OPEN	:
00000000G	00	0000'	CF	PUSHAB	CONV\$AB_RFA_RAB	: 0641
00000000G	00	01	FB	CALLS	#1, SYS\$CONNECT	:
00000000G	CF	10	88	BISB2	#16, CONV\$AB_FLAGS+2	: 0642
00000000G	50	01	D0	MOVL	#1, R0	: 0644
		04	BA	POPR	#^M<R2>	: 0645
		05	00103	RSB		:

4\$:

; Routine Size: 260 bytes, Routine Base: _CONV\$CODE + 0233


```

652 0646 1 %SBTTL 'SET_UP_SORT'
653 0647 1 ROUTINE SET_UP_SORT ( S_TYPE ) : CL$JSB_REG_11 NOVALUE =
654 0648 1 ++
655 0649 1
656 0650 1 Functional Description:
657 0651 1
658 0652 1     Initializes the control blocks for the sort utility
659 0653 1
660 0654 1 Calling Sequence:
661 0655 1
662 0656 1     SET_UP_SORT( sort_type )
663 0657 1
664 0658 1 Input Parameters:
665 0659 1
666 0660 1     sort_type - The sort code for the type of sort wanted. Valid
667 0661 1                  codes are:
668 0662 1                      SOR$GK_RECORD = Record sort (Primary key from non-
669 0663 1                      disk device or multiple input files)
670 0664 1                      SOR$GK_ADDRESS = Rfa sort (Primary key form disk)
671 0665 1                      SOR$GK_INDEX = Index sort (Secondary keys only)
672 0666 1
673 0667 1 Implicit Inputs:
674 0668 1
675 0669 1     KEY_DESC
676 0670 1
677 0671 1 Output Parameters:
678 0672 1     none
679 0673 1
680 0674 1 Implicit Outputs:
681 0675 1     none
682 0676 1
683 0677 1 Routines Called:
684 0678 1
685 0679 1     SOR$BEGIN_SORT
686 0680 1
687 0681 1 Routine Value:
688 0682 1
689 0683 1     Success of error from sor$begin_sort
690 0684 1
691 0685 1 Side Effects:
692 0686 1     none
693 0687 1
694 0688 1 --
695 0689 1
696 0690 2 BEGIN
697 0691 2
698 0692 2 DEFINE_KEY_DESC;
699 0693 2
700 0694 2 ! Sort parameters
701 0695 2 !
702 0696 2 OWN
703 0697 2     KEY_BUFFER      : VECTOR [ 33,WORD ],
704 0698 2     LRL              : WORD,
705 0699 2     SORT_OPTIONS    : LONG,
706 0700 2     SORT_TYPE       : BYTE,
707 0701 2     WORK_FILES      : BYTE;
708 0702 2
```

```

709 0703 2 BIND
710 0704 SEGMENTS = KEY_BUFFER [ 0 ] : WORD,
711 0705 SORT_KEY = KEY_BUFFER [ 1 ] : BLOCKVECTOR [ 8,4,WORD ];
712 0706
713 0707 LOCAL
714 0708 KEY_TYPE;
715 0709
716 0710 SORT_TYPE = .S_TYPE;
717 0711 WORK_FILES = .CONV$GL_WORK_F;
718 0712 LRL = .CONV$GW_MAX_REC_SIZ;
719 0713
720 0714 ! If the key allows dups do a stable sort
721 0715 !
722 0716 IF .KEY_DESC [ KEYSV_DUPKEYS ]
723 0717 THEN
724 0718 SORT_OPTIONS = SORS$M_STABLE
725 0719 ELSE
726 0720 SORT_OPTIONS = _CLEAR;
727 0721
728 0722 ! Get the number of segments
729 0723 !
730 0724 SEGMENTS = .KEY_DESC [ KEYSB_SEGMENTS ];
731 0725
732 0726 ! Find the key type from the key descriptor and set key_type to the
733 0727 ! appropriate SORT-32 code
734 0728 !
735 0729 KEY_TYPE = ( SELECTONE .KEY_DESC [ KEYSB_DATATYPE ] OF
736 0730 SET
737 0731 [ KEYS$C_STRING ] : DSC$K_DTYPE_T;
738 0732 [ KEYS$C_SGNWORD ] : DSC$K_DTYPE_W;
739 0733 [ KEYS$C_SGNLONG ] : DSC$K_DTYPE_L;
740 0734 [ KEYS$C_SGNQUAD ] : DSC$K_DTYPE_Q;
741 0735 [ KEYS$C_UNSGNWORD ] : DSC$K_DTYPE_WU;
742 0736 [ KEYS$C_UNSGNLONG ] : DSC$K_DTYPE_LU;
743 0737 [ KEYS$C_UNSGNQUAD ] : DSC$K_DTYPE_QU;
744 0738 [ KEYS$C_PACKED ] : DSC$K_DTYPE_P;
745 0739 TES );
746 0740
747 0741 ! Load the sort parameter block with the right stuff for each segment
748 0742 !
749 0743 INCR I FROM 0 TO ( .SEGMENTS - 1 ) BY 1
750 0744 DO
751 0745 BEGIN
752 0746 SORT_KEY [ .I,SORTKEY$W_TYPE ] = .KEY_TYPE;
753 0747 SORT_KEY [ .I,SORTKEY$W_ORDER ] = 0;
754 0748
755 0749 ! NOTE: The 28 is the offset to the first segment position descriptor
756 0750 ! in the key descriptor block the 44 is the offset to the segment
757 0751 ! size. If the macros for these ever change, ie. KEYS$W_POSITION and
758 0752 ! KEYS$B_SIZE, the code offsets here must be changed!
759 0753 !
760 0754 SORT_KEY [ .I,SORTKEY$W_START ] = .KEY_DESC [ ( 28 + (.I*2) ),WORD_U ];
761 0755 SORT_KEY [ .I,SORTKEY$W_LENGTH ] = .KEY_DESC [ ( 44 + .I ),BYTE_U ];
762 0756
763 0757 ! If the key is packed decimal then sort wants the size in nibbles NOT
764 0758 ! counting the sign
765 0759 !
```



```

: 766      0760 3      IF .KEY_DESC [ KEYSB_DATATYPE ] EQLU KEYSB_PACKED
: 767      0761      THEN
: 768      0762          SORT_KEY [ .I, SORTKEY$W_LENGTH ] =
: 769      0763              ( .SORT_KEY [ .I, SORTKEY$W_LENGTH ] * 2 ) - 1
: 770      0764
: 771      0765      END;
: 772      0766
: 773      0767      ! Begin the sort
: 774      0768      !
: 775      0769      SORS$BEGIN_SORT( KEY_BUFFER,      ! Key buffer address
: 776      0770          LRL,      ! Longest record length
: 777      0771          SORT_OPTIONS,      ! Sort options
: 778      0772          0,      ! Input file size
: 779      0773          0,      ! Comp. routine addr.
: 780      0774          0,      ! Equal routine addr.
: 781      0775          SORT_TYPE,      ! Sort type
: 782      0776          WORK_FILES );      ! Number of work files
: 783      0777
: 784      0778      RETURN
: 785      0779
: 786      0780 1      END;
```

.PSECT _CONVSOWN,NOEXE, PIC,2

```
00078 KEY_BUFFER:
      .BLKB 66
000BA LRL:      .BLKB 2
000BC SORT_OPTIONS:
      .BLKB 4
000C0 SORT_TYPE:
      .BLKB 1
000C1 WORK_FILES:
      .BLKB 1
```

```
SEGMENTS=      KEY_BUFFER
SORT_KEY=      KEY_BUFFER+2
```

.PSECT _CONV\$CODE,NOWRT, SHR, PIC,2

```

007C 8F BB 00000 SET_UP_SORT:
0000' CF 18 AE 90 00004      PUSH R2,R3,R4,R5,R6
0000' CF 0000G CF 90 0000A      MOV S_TYPE, SORT_TYPE
0000' CF 0000G CF B0 00011      MOV CONVSGL_WORK_F, WORK_FILES
      OB 10 AB E9 00018      MOV CONVSGL_MAX_REC_SIZ, LRL
0000' CF 00000000G 8F D0 0001C      BLBC 16(KEY_DESC), 1$
      04 11 00025      MOVL #SORS$STABLE, SORT_OPTIONS
      CF D4 00027 1$:      BRB 2$
0000' CF 12 AB 9B 0002B 2$:      CLRL SORT_OPTIONS
      53 11 AB 9A 00031      MOVZBW 18(KEY_DESC), SEGMENTS
      05 12 00035      MOVZBL 17(KEY_DESC), R3
      54 0E D0 00037      BNEQ 3$
      49 11 0003A      MOVL #14, KEY_TYPE
      01 53 91 0003C 3$:      BRB 11$
      CMPB R3, #1
: 0647
: 0710
: 0711
: 0712
: 0716
: 0718
: 0720
: 0724
: 0729
: 0731
: 0732
```

	05	12	0003F	BNEQ	4\$		
54	07	D0	00041	MOVL	#7	KEY_TYPE	
	3F	11	00044	BRB	11\$		
03	53	91	00046	4\$: CMPB	R3, #3		0733
	05	12	00049	BNEQ	5\$		
54	08	D0	0004B	MOVL	#8	KEY_TYPE	
	35	11	0004E	BRB	11\$		
06	53	91	00050	5\$: CMPB	R3, #6		0734
	05	12	00053	BNEQ	6\$		
54	09	D0	00055	MOVL	#9	KEY_TYPE	
	2B	11	00058	BRB	11\$		
02	53	91	0005A	6\$: CMPB	R3, #2		0735
	05	12	0005D	BNEQ	7\$		
54	03	D0	0005F	MOVL	#3	KEY_TYPE	
	21	11	00062	BRB	11\$		
04	53	91	00064	7\$: CMPB	R3, #4		0736
	05	12	00067	BNEQ	8\$		
54	04	D0	00069	MOVL	#4	KEY_TYPE	
	17	11	0006C	BRB	11\$		
07	53	91	0006E	8\$: CMPB	R3, #7		0737
	05	12	00071	BNEQ	9\$		
54	05	D0	00073	MOVL	#5	KEY_TYPE	
	0D	11	00076	BRB	11\$		
05	53	91	00078	9\$: CMPB	R3, #5		0738
	05	13	0007B	BEQL	10\$		
54	01	CE	0007D	MNEGL	#1	KEY_TYPE	
	03	11	00080	BRB	11\$		
54	15	D0	00082	10\$: MOVL	#21	KEY_TYPE	
55	0000'	CF	3C 00085	11\$: MOVZWL	SEGMENTS, R5		0743
50		01	CE 0008A	MNEGL	#1, I		0746
		34	11 0008D	BRB	13\$		
	0000'CF	40	7F 0008F	12\$: PUSHAQ	SORT_KEY[I]		
9E		54	B0 00094	MOVW	KEY_TYPE, @ (SP)+		
	0000'CF	40	7F 00097	PUSHAQ	SORT_KEY+2[I]		0747
		9E	B4 0009C	CLRW	@ (SP)+		
	0000'CF	40	7F 0009E	PUSHAQ	SORT_KEY+4[I]		0754
9E	1C AB	40	B0 000A3	MOVW	28(KEY_DESC)[I], @ (SP)+		
51	0000'CF	40	7E 000A8	MOVAQ	SORT_KEY+6[I], R1		0755
61	2C A0	4B	9B 000AE	MOVZBW	44(I)[KEY_DESC], (R1)		
05		53	91 000B3	CMPB	R3, #5		0760
		0B	12 000B6	BNEQ	13\$		
52		61	3C 000B8	MOVZWL	(R1), R2		0763
52		01	78 000BB	ASHL	#1, R2, R6		
56		01	A3 000BF	SUBW3	#1, R6, (R1)		
50		55	F2 000C3	13\$: AOBLS	R5, I, 12\$		0760
	0000'	CF	9F 000C7	PUSHAB	WORK_FILES		0769
	0000'	CF	9F 000CB	PUSHAB	SORT_TYPE		
		7E	7C 000CF	CLRW	-(SP)		
		7E	D4 000D1	CLRL	-(SP)		
	0000'	CF	9F 000D3	PUSHAB	SORT_OPTIONS		
	0000'	CF	9F 000D7	PUSHAB	LRL		
	0000'	CF	9F 000DB	PUSHAB	KEY_BUFFER		
	00000000G	00	08 FB 000DF	CALLS	#8, -SOR\$BEGIN_SORT		
		007C	8F BA 000E6	POPR	#M<R2,R3,R4,R5,R6>		0780
		05	000EA	RSB			

; Routine Size: 235 bytes, Routine Base: _CONV\$CODE + 0337

CONVSORT
V04-000

VAX-11 CONVERT
SET_UP_SORT

M 8
15-Sep-1984 23:48:01
14-Sep-1984 12:14:02

VAX-11 Bliss-32 V4.0-742
[CONV.SRC]CONVSORT.B32;1

Page 25
(7)

: 787
: 788 0781 1
0782 0 END ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
CONVSPLIT	24	NOVEC,NOWRT, RD,NOEXE, SHR, LCL, REL, CON, PIC,ALIGN(2)
CONVSOWN	194	NOVEC, WRT, RD,NOEXE,NOSHR, LCL, REL, CON, PIC,ALIGN(2)
CONV\$GLOBAL	152	NOVEC, WRT, RD,NOEXE,NOSHR, LCL, REL, CON, PIC,ALIGN(2)
CONV\$CODE	1058	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	120	0	1000	00:01.8
_\$255\$DUA28:[CONV.SRC]CONVERT.L32;1	165	27	16	17	00:00.2

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:CONVSORT/OBJ=OBJ\$:CONVSORT MSRC\$:CONVSORT/UPDATE=(ENH\$:CONVSORT)

: Size: 1058 code + 370 data bytes
: Run Time: 00:22.2
: Elapsed Time: 01:16.0
: Lines/CPU Min: 2112
: Lexemes/CPU-Min: 26082
: Memory Used: 183 pages
: Compilation Complete

0066 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

